Manual Técnico

Constantes..... pag 2

Rangos..... pag 2

Registros..... .....pag 2

Arrays..... pag 3

Gastos y expensas .....pag 4

AMB edificio_____

Alta edificios..... pag 11

Modificacion edificios..... pag 17

Baja edificios..... pag 19

AMB Departamentos_____

Alta departamentos..... pag 20

Modificacion departamentos..... pag 21

Baja departamentos..... pag 22

Carga de prueba..... pag 23

Encargados .....pag 25

Modelo de datos:

Para la carga de gastos se utiliza una matriz de días (CONSTANTE: MAXDIAS) x meses (CONSTANTE: MAXMES). Cada posición de esta matriz está compuesta por el registro trGastos. Este posee en sus campos, el listado de gastos fijos, el listado de gastos extraordinarios. El total de gastos fijos, el total de gastos extraordinarios y el total de los gastos, que es la suma de los gastos fijos más los gastos extraordinarios.

Constantes

MINMES: Constante que es utilizada para contener el número mínimo de meses.

MINMES: Constante que es utilizada para contener el número máximo de meses.

MINANIO: Constante que es utilizada para contener el número mínimo de año a utilizar.

MAXANIO: Constante que es utilizada para contener el número máximo de año a utilizar.

MINGASTOS: Constante que es utilizada para contener el número mínimo de gastos.

MAXGASTOS: Constante que es utilizada para contener el número máximo de gastos.

Rangos

tranMes: Rango que se encuentra entre las constantes **MINMES** y **MAXMES**

tranAnio: Rango que se encuentra entre las constantes **MINANIO** y **MAXANIO**

tranNumGastos: Rango que se encuentra entre las constantes **MINGASTOS** y **MAXGASTOS**

Registros

trGastosConcepto: Registro con 2 campos, gasto y concepto. En el campo gasto (real) se encuentra el monto del gasto. En el campo concepto (string [40]) se encuentra la descripción del gasto.

trGastos: Registro con 6 campos. A continuación se hará una breve descripción de cada campo.

SueldoEncargado: Campo en el que se encuentra el valor (real) del sueldo del encargado.

listadoGastosFijos: Campo en el que se encuentra el vector de gastos fijos (tv_Gastos).

listadoGastosExtraordinarios: Campo en el que se encuentra el vector de gastos fijos (tv_Gastos).

totalGastosFijos: Campo en el que se encuentra el valor (real) del total de gastos fijos.

totalGastosExtra: Campo en el que se encuentra el valor (real) del total de gastos extraordinarios.

gastosTotales:. Campo en el que se encuentra el valor (real) del total de gastos esto es la suma de gastos fijos y los gastos totales.

trGastosExpensas: Registro con 3 campos. Este registro se usa para diferenciar las expensas fijas de las

extraordinarias, ya que las expensas fijas serán pagadas por los dueños y/o inquilinos y las extraordinarias son pagadas solo por los dueños. A continuación se hará una breve descripción de cada campo.

totalGastosFijos: Campo en el que se encuentra el valor (real) del total de gastos fijos.

totalGastosExtra: Campo en el que se encuentra el valor (real) del total de gastos extraordinarios.

gastosTotales: Campo en el que se encuentra el valor (real) del total de gastos esto es la suma de gastos fijos y los gastos totales.

Arrays

tv_Gastos: Vector de rango tranNumGastos, en cada posición del vector se encuentra el registro trGastosConcepto. En este vector se cargara el listado de gastos

tmGastosMesAnio: Matriz de rangos tranMes, tranAnio. En cada posición de la matriz se encuentra el registro trGastos. En esta matriz se cargaran los gastos del edificio para distintas fechas (mes/año).

tmExpensas: Matriz de rangos traneEdificio, tranPisos, tranDtos, tranMes, tranAnio. En cada posición de la matriz se encuentra un registro del tipo trGastosExpensas. Esta matriz será en donde se encontrara el manejo de expensas de los edificios.

Gatos y expensas

calcularTotalGastoExtra: Esta función recibe como parámetros la matriz edificios, el código de edificio, el mes y el año a calcular el total de gastos extraordinarios. Mediante la implementación de un ciclo for, se acumula en una variable local el listado de gastos extraordinarios del edificio. El ciclo está limitado en su tope de iteraciones por el valor de la cantidad de gastos extraordinarios que tenga ese edificio. Al terminar el ciclo el valor de la variable encargada de acumular se le asigna a la función.

listarGastosExtra: Este procedimiento recibe como parámetros el registro Administración, el vector edificios, y el mes y el año para el cual se desea listar los gastos extraordinarios. Mediante un ciclo repeat se encarga de validar que el código de edificio ingresado se un valor valido. Una vez ingresado el código del edificio se recorrerá el listado de gastos extraordinarios con un ciclo for, desde la constante MINGASTOS hasta la cantidad de gastos extraordinarios en este momento.

cargarSueldoEncargadoEnMes: Este procedimiento recibe como parámetros el registro administración por valor y el vector encargados y el vector edificios por referencia. En el procedimiento se utilizan varios ciclos for para ir cargando en una variable local el monto de los sueldos de los encargados que correspondan al edificio para cada mes. Una vez cargada la variable local para ese mes, al vector edificio en su matriz gastos para un mes y un determinado año, en el campo sueldo encargado, se le es asignado el valor al edificio en que estamos posicionados.

cargaGastosExtra: A este procedimiento se le pasan como parámetros el vector edificios, el código del edificio al, el mes y el año al cual le estoy cargando los gastos. Este procedimiento contiene un ciclo while que por condición pide que el usuario desea seguir ingresando gastos y que no me haya ido del máximo de gastos, dado por la constante MAXGASTOS. Al iniciar el procedimiento se setea la variable local continuarIngresando en verdadero y la variable para contar los gastos ingresados en 0 para que automáticamente se ingrese al ciclo. Una vez ingresado al ciclo se incrementa la cantidad de gastos extraordinarios del edificio y la variable para contar los gastos ingresados en ese momento, se imprimen por pantalla las instrucciones y luego se procede pidiendo el concepto del gasto (cadena de caracteres) y mediante un ciclo repeat se valida que el dato ingresado al solicitar el costo sea un entero. Al finalizar la carga de datos se pregunta si desea seguir continuando mediante la función Pregunta. Si el usuario desea seguir ingresando gastos ingresa el carácter ‘s’ o ‘S’ y la variable continuarIngresando se mantiene en true, en caso de que el usuario ingrese ‘n’ o ’N’ continuarIngresando cambia su estado a false y sale del ciclo.

GastosExtra; Este procedimiento es el encargado de presentar el menú de carga de gastos extraordinarios. Recibe como parámetros el registro administración y por referencia el vector edificios. Al comenzar el procedimiento se valida que por lo menos exista un edificio cargado, si esto sucede se despliega la lista de edificios con sus códigos mediante el procedimiento listaEdificios, en caso contrario se imprime por pantalla un mensaje que indica que todavía no existen edificios cargados. Luego pide el ingreso del código del edificio al cual se desea que le carguen los gastos. Una vez ingresado el código del edificio se pregunta si desea cargar los gastos para la fecha actual del sistema o para una fecha posterior. En caso de elegir la fecha actual, se cargan los gastos extraordinarios para el mes y año actual y luego se calcula el total de gastos extraordinarios para la fecha del sistema. En caso elegir una fecha posterior, se valida la entrada del mes cumpliendo que sea un dato valido y posterior o igual a la fecha del sistema, después se cargan los gastos extraordinarios para el mes elegido y luego se calcula el total de gastos extraordinarios para dicho mes.

listarGastosFijos: Este procedimiento recibe como parámetros el registro Administración, el vector edificios, y el mes y el año para el cual se desea listar los gastos fijos. Mediante un ciclo repeat se encarga de validar que el código de edificio ingresado se un valor valido. Una vez ingresado el código del edificio se recorrerá el listado de gastos fijos con un ciclo for, desde la constante MINGASTOS hasta la cantidad de gastos fijos en este momento.

calcularTotalGastoFijos: Esta función recibe como parámetros la matriz edificios, el código de edificio, el mes y el año a calcular el total de gastos fijos. Mediante la implementación de un ciclo for, se acumula en una variable local el listado de gastos fijos del edificio. El ciclo está limitado en su tope de iteraciones por el valor de la cantidad de gastos fijos que tenga ese edificio. Al terminar el ciclo el valor de la variable encargada de acumular se le asigna a la función.

cargaGastosFijos: A este procedimiento se le pasan como parámetros el vector edificios, el código del edificio al, el mes y el año al cual le estoy cargando los gastos. Este procedimiento contiene un ciclo while que por condición pide que el usuario desea seguir ingresando gastos y que no me haya ido del máximo de gastos, dado por la constante MAXGASTOS. Al iniciar el procedimiento se setea la variable local continuarIngresando en verdadero y la variable para contar los gastos ingresados en 0 para que automáticamente se ingrese al ciclo. Una vez ingresado al ciclo se incrementa la cantidad de gastos fijos

del edificio y la variable para contar los gastos ingresados en ese momento, se imprimen por pantalla las instrucciones y luego se procede pidiendo el concepto del gasto (cadena de caracteres) y mediante un ciclo repeat se valida que el dato ingresado al solicitar el costo sea un entero. Al finalizar la carga de datos se pregunta si desea seguir continuando mediante la función Pregunta. Si el usuario desea seguir ingresando gastos ingresa el carácter 's' o 'S' y la variable continuarIngresando se mantiene en true, en caso de que el usuario ingrese 'n' o 'N' continuarIngresando cambia su estado a false y sale del ciclo.

GastosFijos; Este procedimiento es el encargado de presentar el menú de carga de gastos fijos. Recibe como parámetros el registro administración y por referencia el vector edificios. Al comenzar el procedimiento se valida que por lo menos exista un edificio cargado, si esto sucede se despliega la lista de edificios con sus códigos mediante el procedimiento listaEdificios, en caso contrario se imprime por pantalla un mensaje que indica que todavía no existen edificios cargados. Luego pide el ingreso del código del edificio al cual se desea que le carguen los gastos. Una vez ingresado el código del edificio se pregunta si desea cargar los gastos para la fecha actual del sistema o para una fecha posterior. En caso de elegir la fecha actual, carga en la primer y segunda posición del listado de gastos fijos, el sueldo del encargado y los gastos de administración (en el campo concepto se cargan las variables **SUELDOENCARGADO** y **GASTOSADMINISTRACION**), luego se cargan los gastos fijos para el mes y año actual y luego se calcula el total de gastos fijos para la fecha del sistema. En caso elegir una fecha posterior, se valida la entrada del mes cumpliendo que sea un dato valido y posterior o igual a la fecha del sistema, después carga en la primer y segunda posición del listado de gastos fijos, el sueldo del encargado y los gastos de administración y luego se cargan los gastos fijos para el mes elegido y luego se calcula el total de gastos fijos para dicho mes.

ListadoGastos: Este procedimiento se encarga del menú listados de gastos. Al comenzar el procedimiento se valida la existencia de al menos un edificio cargado en la administración, en caso de que esto no ocurra se imprime por pantalla un mensaje que indica que todavía no existen edificios cargados. Caso contrario se prosigue y entra en el ciclo while que monitorea si el usuario desea continuar en ese menú. A continuación pregunta si desea mostrar los listados para la fecha actual del sistema o para fechas posteriores. Si la opción elegida mediante la función Pregunta es la fecha actual, se imprime por pantalla si desea listar los gastos fijos o los extraordinarios. Luego mediante un case se ejecuta el procedimiento listarGastosFijos o listarGastosExtra, dependiendo de la opción elegida, pasándole como parámetro el mes y el año actual del sistema. Si la opción elegida mediante la función Pregunta no es la fecha actual, se imprime por pantalla si desea listar los gastos fijos o los extraordinarios. Luego mediante un case se ejecuta el procedimiento listarGastosFijos o listarGastosExtra, dependiendo de la opción elegida, pasándole como parámetro el mes ingresado por el usuario.Si la opción elegida es volver la variable que utiliza el while como condición cambia a falso y se sale del ciclo, volviendo al menú anterior.

CargarGastos: Este procedimiento se encarga del menú carga de gastos. Al comenzar el procedimiento se valida la existencia de al menos un edificio cargado en la administración, en caso de que esto no ocurra se imprime por pantalla un mensaje que indica que todavía no existen edificios cargados. Caso contrario se prosigue y entra en el ciclo while que monitorea si el usuario desea continuar en ese menú. A continuación se ejecuta el procedimiento cargarSueldoEncargadoEnMes que ya fue mencionado anteriormente. Luego se imprime en pantalla las opciones del menú, en caso de elegir la opción 0, la variable que utiliza el while como condición cambia a falso y se sale del ciclo, volviendo al menú anterior. En caso de que la opción sea 1 se ejecuta el procedimiento GastosFijos, en caso de que sea 2 se ejecuta el procedimiento GastosExtra.

Gastos: Este procedimiento se encarga del menú de gastos. Al comenzar el procedimiento se valida la existencia de al menos un edificio cargado en la administración, en caso de que esto no ocurra se imprime por pantalla un mensaje que indica que todavía no existen edificios cargados. Caso contrario se prosigue y entra en el ciclo while que monitorea si el usuario desea continuar en ese menú. Luego se imprime en pantalla las opciones del menú, en caso de elegir la opción 0, la variable que utiliza el while como condición cambia a falso y se sale del ciclo, volviendo al menú anterior. En caso de que la opción sea 1 se ejecuta el procedimiento CargarGastos, en caso de que sea 2 se ejecuta el procedimiento ListarGastos.

ValidarPagoExpensasMes: Procedimiento que recibe por parámetro la matriz edificios, el código del edificio, el piso, el departamento, el mes y por referencia la variable deudaVieja. Luego valida que el mes del último pago del departamento sea menor que el mes en que desea cargar las expensas. Si el mes del último pago es menor entonces asigno a deudaVieja true, caso contrario le asigno false. Si el estado de deudaVieja es false entonces al día del último pago del departamento le asigno la cantidad de días que posee el mes mediante la función **cantDiasMes.**

calculoExpensas: Esta función calcula el total de las expensas para ese departamento. Para lograr este cálculo primero obtiene el valor del total de las expensas fijas y extraordinarias. El valor del total de las expensas de ese departamento es la suma del total de las expensas fijas, las extraordinarias y el monto de la multa por mora calculado por la función **calculosaldoanterior**. Luego muestra por pantalla los datos de las expensas para ese departamento y otros datos complementarios.

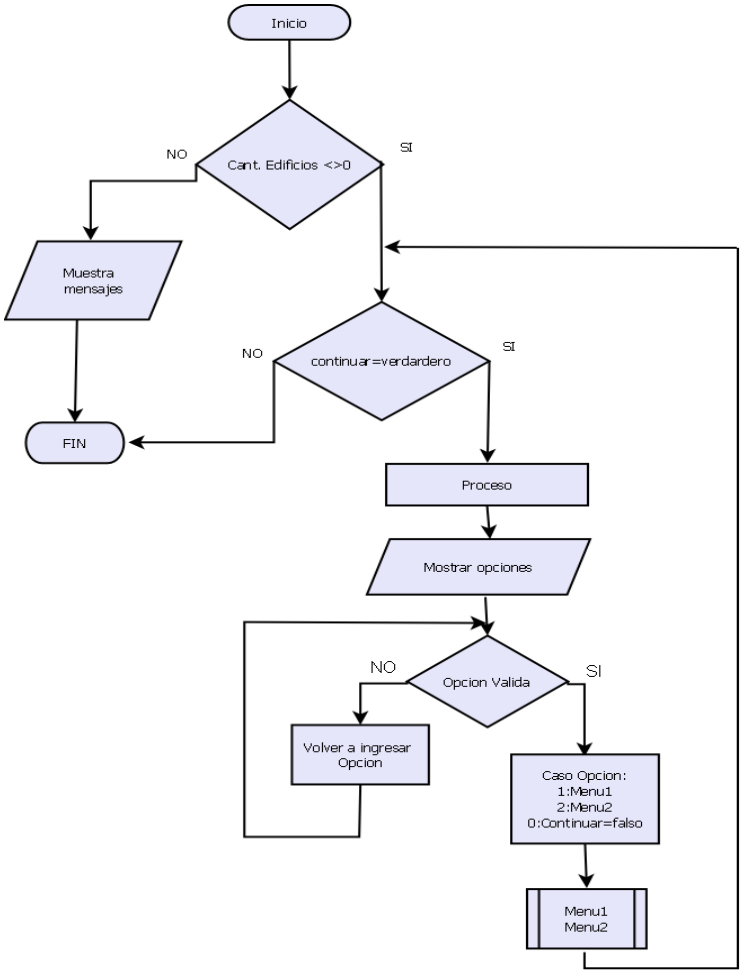
ExpensasEdificio: Este procedimiento se encarga de calcular las expensas para los departamentos habitados del edificio. Al iniciar el procedimiento valida la existencia de edificios en la administración, luego se despliega el listado de edificios con sus códigos para poder ingresar el código de edificio que posteriormente es validado mediante un ciclo de validación de datos ingresados. Una vez validado el

código del edificio se procede preguntando si desea calcular las expensas para el mes actual o para meses posteriores del sistema. Se guardan en variables locales los totales de los gastos extraordinarios y los gastos fijos. Así mismo se calculan los puntos totales del edificio mediante la función **puntostotales** y el valor se le asigna a una variable local. Si el total de gastos fijos es cero entonces no se podrán calcular las expensas del edificio, mostrando un mensaje con esta consigna. Caso contrario se recorren los pisos y los departamentos del edificio mediante ciclos for anidados. Por cada departamento se valida si está habilitado o no. Luego valida que la fecha del último pago del departamento sea menor que la fecha en que desea cargar las expensas. Si la fecha del último pago es menor entonces se entra en el ciclo que calcula lanza los procedimientos **ValidarPagoExpensasMes** y **calculoExpensas .**

CargarExpensa: Este procedimiento es el encargado de registrar el pago que realiza un departamento (comprende a el pago efectuado por el inquilino + el dueño). Al comenzar el usuario verá en pantalla, gracias a la funcion *pregunta(6)* si la fecha de pago es la actual. Si es asi, por medio de un if se cargara automáticamente la fecha del dia actual en los campos de diadelultimopago. Luego se pedirá el monto de dinero que fue pagado.

Caso contrario, es decir , la fecha actual no es en la cual se registró el pago. El usurio deberá ingresar dia, mes y año en el cual se realizó el pago para luego guardar esos datos en diadelultimopago. A continuación le pedirá el monto de dinero para el pago.

Ejemplo de diagrama de flujo de carga de gastos extraordinarios y carga de gastos



Ejemplo de pseudocódigo de procedimiento listadoGastos

```
Procedimiento ListadoGastos(Parametros);

Variables
Declaro variables locales

Inicio del procedimiento
        Si(hay algun edificio cargado) entonces
                Mientras (deseo continuar listando gastos) hacer

                        Mostrar Menu Por Pantalla
                        Pregunto si deseo utilizar fecha del sistema o posteriores

                        Si (elijo fecha Actual del sistema) entonces

                                Mostrar Menu Por Pantalla
                                Validar Opcion
                                Ejecutar Case
                        Sino

                                Validar Mes y Año
                                Mostrar Menu Por Pantalla
                                Validar Opcion
                                Ejecutar Case
                        Fin del si
                Fin del continuar;
        Sino
                Mostrar mensaje de que no existen edificios cargados
        Fin del si
Fin del procedimiento
```

```
            val(opcion,valor,codigo);
until (codigo=0) and (valor>=MINIMO) and (valor<=MAXIMO);
```

ALTAS EDIFICIOS

El procedimiento encargado de las altas de edificios es el siguiente:

```
PROCEDURE cargarEdificios(var Administracion:trAdministracion;var edificios:tvEdificio;var
encargados:t_vencargados);
```

Los parámetros que utiliza este procedimiento son utilizados para lo sig:

- Administración: Utilizamos el campo "cantedificios" para ir cargando la cantidad de edificios que tendremos y principalmente para validar que no nos hemos pasado de la cantidad máxima de edificios que establecimos anteriormente

while((Administracion.cantEdificios<= MAXEDIFICIOS) AND continuar) do

{continuar es el flag que utilizamos para volver a cargar otro edificio si es pedido por el usuario. Inicialmente puesto en true.}

-Edificios: En este vector cargaremos todos los datos de los edificios (todos los registros que posee dentro dicho vector)

-Encargados: Con este parámetro mostraremos a través de otro procedimiento el nombre del encargado que corresponde a un determinado edificio.

El funcionamiento del procedimiento se basa en un ciclo *while* con dos condiciones

1. Administracion.cantEdificios<= MAXEDIFICIOS

2. continuar= true.

La primera me asegura no cargar más edificios de los que puedo. La segunda será false si el usuario no desea cargar mas edificios.

Luego, antes de pedir los datos del edificio para cargarlos en sus respectivos registros, nos aseguramos que el edificio no exista.

if(busquedaEdificio(edificios,Administracion.cantEdificios,Administracion.cantEdificios) =0)

Si la función *BusquedaEdificio* devuelve un cero, significa que el edificio no existe. Se puede comenzar con la carga de datos.

El pedido de ingreso de datos es el mismo para todos los datos, solo cambia en donde se guardan. En este caso mostramos como lo hacemos para la antigüedad del edificio.

```
repeat
        begin
        write('Ingrese la antigüedad del edificio en anios: ');
        readln(antigüedad);
                        val(antiguedad,edificios[Administracion.cantEdificios].antiguedad,codigo);

        if(código <>0)then writeln('Debe ingresar un numero...');
      end;
                                        until(((edificios[Administracion.cantEdificios].antiguedad)>=0)
and (codigo=0)) ;
```

Como hemos hecho en todo el programa, para validar usamos la misma estructura. La función val y luego la condición del repeat para terminar de validar el ingreso de datos.

A continuación, después de la carga de datos (hasta puntos para los departamentos), se llaman a los sig procedimientos

inicializarPisos(edificios,Administracion.cantEdificios);

Pone en cero el registro cantdeptosporpiso. También pone en false el registro "especial" que es el flag que usamos para determinar si es un depto especial o no.

completarDepartamentosEspeciales(edificios,Administracion.cantEdificios);

A este procedimiento se le pasa como parámetros "edificios " y "numedificio".

Numedificio es el edificio con el que se viene trabajando (al cual se le están cargando los datos).

Primero le pide al usuario que ingrese que piso desea cargar. Una vez ingresado el dato y si es valido se despliega un menú para la carga de puntos con las siguientes opciones

1.lujoso

edificios[numEdificio].cantdptosPorPiso[piso].CantDptos:=2;

edificios[numEdificio].cantdptosPorPiso[piso].especial:=true;

puntos:=10;

2.Básico

edificios[numEdificio].cantdptosPorPiso[piso].CantDptos:=edificios[numEdificio].cantMaxDptosPorPiso;

edificios[numEdificio].cantdptosPorPiso[piso].especial:=true;

puntos:=9;

3.Extra lujoso

edificios[numEdificio].cantdptosPorPiso[piso].CantDptos:=1;

edificios[numEdificio].cantdptosPorPiso[piso].especial:=true;

puntos:=20;

{Solo son datos precargados. El usuario puede ingresarlos manualmente si lo desea.}

4.carga manual

En esta carga, primero se pide ingresar el numero del piso para luego, por medio de esta línea "edificios[numEdificio].cantdptosPorPiso[piso].especial:=true; " habilitar los departamentos.

Eventualmente se pide el ingreso de los puntos para los departamentos de ese piso.

Finalmente se carga en el campo *numDepartamento* el numero del departamento con el siguiente formato (1-"A")

completarDepartamentosEstandar(edificios,Administracion.cantEdificios);

Este es el procedimiento que se encarga se asignarle los "nombres" (1-"A") a los departamentos y los puntos previamente ingresados.

Luego se coloca en true el campo *habilitado* para ese edificio. Usaremos bajas lógicas para los edificios.

Seguido comienza la carga de departamentos (si es que el usuario lo desea. Caso contrario lo podrá hacer desde las modificaciones de los deptos). La carga de departamentos será tratada mas adelante.

Finalmente, luego de cargar todo el edificio, el usuario, a través del procedimiento *MostrarDatosEdificio* verá los datos ingresados, si estos datos son incorrectos los podrá modificar mediante la llamada al procedimiento

"

ModificarCampos(administracion,edificios,Administracion.cantEdificios,B);

"

{La letra b sirve de flag para indicar que menú de modificar campos tiene que dirigirse. Si la opción modificar edificios es llamada desde aquí, tendrá otras opciones que si es llamada desde el menú ABM edificios. El usuario puede equivocarse al ingresar datos con respecto a los departamentos y pisos, pero una vez ingresados no.

los podrá modificar desde el ABM edificios dado que la cantidad de pisos o departamentos en un edificio no cambia.}

MODIFICACION EDIFICIOS

Desde el menú de ABM de edificios se llama al procedimiento *ElegirEdificioModificar.*

Se le pasan los parámetros *administración* y *edificios.* En el, primero, se muestra la lista de edificios que existen para que el usuario elija cual quiere modificar. Una vez ingresado el edificio, y que exista, es decir, que su campo "habilitado" sea true, se llama al procedimiento *ModificarCampos.*

El procedimiento *ModificarCampos* despliega un menú según de donde sea llamado. Esto se debe a un flag "B" que si es 1 despliega un menú y si es 0 despliega otro. Como se ha comentado antes, esto se debe a la posibilidad del error de ingreso de datos del usuario.

Si es llamado desde AMB edificios (B=0) el menú es el siguiente

1)Nombre

2)Dirección

3)Ascensores

4)Antiguedad

5)Multa por mora

0)Volver al menú anterior

Se trata de un case que según la opción ingresada, le pide al usuario los nuevos datos para modificar los campos, nombre,direccion,antigüedad etc.

Si es llamado desde *cargaEdficios* (B=1):

1)Nombre

2)Dirección

3)Pisos/Departamentos

4)Ascensores

5)Antiguedad

6)Multa por mora

Presione "0" para salir

Se trata de otro case pero con la opción 3 que no aparece en el menú llamado desde ABM.

BAJA EDIFICIOS

Hemos decidido hacer bajar lógicas. Lo cual hace muy simple esta parte del programa .

El procedimiento *elegirbajaedificio* se encarga de seleccionar el edificio para luego llamar a *bajaedificio* el cual pone en “false” al campo *habilitado* que se encuentra en cada edificio.

Colocamos la opción recuperar edificio, el cual pone *habilitado* en true, en caso de que el usuario se equivoque al dar de baja un edificio.

ALTAS DEPARTAMENTOS

En el caso de las altas de departamentos, las pusimos seguido de la carga de edificios (dentro del procedimiento *cargaredificios*, el cual se encarga de las altas de edificios). Si el usuario lo desea puede cargar desde modificación de departamentos.

El comienzo de las altas de departamentos comienza en la línea 2268, donde se inicializan todos los departamentos. A continuación se pide el ingreso de cual departamento se desea cargar. Por medio de la función *busquedadeptoacargar* ratifica la existencia del mismo y si existe, llama al procedimiento *cargardeptos.*

Dicho procedimiento, *cargardeptos,* se le pasa como parámetros la administración, edificios, el edificio sobre el cual se trabajará (numedificio), el piso y el departamento. La primer pregunta que hace el procedimiento es para saber si es dueño o inquilino el habitante que allí vive, debido a que según su respuesta serán distintos los campos a cargar. Si es dueño, debe cargar solo nombre y seguido se carga el día del ultimo pago con la fecha actual. En cambio si es inquilino debe cargar el nombre del dueño, donde reside y el nombre del inquilino que esta habitando ese departamento. Posteriormente, como en el caso anterior, también se carga la fecha del último pago como la fecha actual.

Para finalizar coloca en true el flag *habilitado* que será usado para las bajas lógicas.

MODIFICACION DEPARTAMENTOS

El procedimiento encargado de realizar la modificación de departamentos es el sig:

```
procedure modificardeptos(var administracion:tradministracion;var
edificios:tvEdificio;edificioelegido:byte;deptoamodificar:tnumdepartamento;piso:tcantpisos;departamen
to:tcantmaxdptosporpiso);
```

Como se ve, los parámetros que usa este procedimiento son: Administración, edificios, edificioelegido (sobre el cual se trabajará), deptoamodificar, piso, departamento.

El parámetro deptoamodificar viene del procedimiento (*elegirdeptoamodificar)* y como lo dice su nombre, es el departamento que se modificará.

El procedimiento *modificardeptos* consta de las sig opciones:

1) Dueño o Inquilino?

2)Datos del dueño

3)Datos del inquilino

4)Fecha del último pago

0)Volver al menú anterior

Las primeras opciones (1,2 y 3) solo son cambios en los campos de los departamentos (nombre, dirección, D/ I). En cambio la opción 4 es la que indica, grabando los datos ingresados en fechadelultimopago(campo en departamentos), cuando se registró el

Ultimo pago para ese departamento. De esta manera cobraremos la multa que corresponda por mora.

BAJA DEPARTAMENTOS

El procedimiento encargado de la baja de departamentos es el sig:

```
procedure eliminardepartamento(var edificios:tvEdificio;
edificioelegido:tcodedificio;numdepartamento:tnumdepartamento);
```

tiene como parámetros : administración, edificios, edificioelegido,numdepartamento.

Vale la pena aclarar que el parámetro numdepartamento es un string y tiene adentro información del tipo (1-“A” por ejemplo). De esta manera se invoca a una búsqueda binaria para asegurar la existencia del departamento. Si el departamento se encontraba habilitado, entonces se pondrá en false el campo *habilitado* dando así una baja lógica.

CARGA DE DATOS

El procedimiento *Datosdepruebaedificio* es el encargado por medio de un for, de cargar los datos de prueba para los edificios. Luego el procedimiento *cargadeptosprueba* se encarga de completar la información de los departamentos para cada edificio. La estructura es similar a la de *datospruebaedificio*, un for para los 3 edificios cargados

Los edificios cargados son los siguientes :

Edificio 1:

Nombre: Edificio Cavanagh

Dirección: Av. Paseo Colón 850

Antigüedad: 3

Ascensores: 2

Cantidad de pisos: 3

Cantidad de departamentos estándar por piso: 3

Cantidad de departamentos en planta baja: 1

Cantidad de departamentos en el último piso: 2

Multa impuesta por el edificio en caso de mora: 10

Edificio 2:

Nombre: Torre de los Fresnos

Dirección: 9 de julio 666

Antigüedad: 5

Ascensores: 3

Cantidad de pisos: 4

Cantidad de departamentos estándar por piso: 4

Cantidad de departamentos en planta baja: 2

Cantidad de departamentos en el último piso: 3

Multa impuesta por el edificio en caso de mora: 15

Edificio 3:

Nombre: Soho

Dirección: av ing. huergo 450

Antigüedad: 5

Ascensores: 4

Cantidad de pisos: 5

Cantidad de departamentos estándar por piso: 5

Cantidad de departamentos en planta baja: 3

Cantidad de departamentos en el último piso: 4

Multa impuesta por el edificio en caso de mora: 1

ENCARGADOS

La seccion “encargados” cuenta con un procedimiento llamado “**MENÚ_ENCARGADOS**” , se inicializan las variables definidas en este procedimiento. Posteriormente se ingresa a un ciclo “while” donde el ciclo deja de ejecutarse si la variable “indicador” toma el valor 0 (cero), el valor 0 (cero) se utiliza para volver al menú anterior.

Al inicializarse en un valor del indicador de 10, se ejecuta el while, el ciclo muestra al comienzo el menú de opciones para los encargados. Y se pide que se ingrese un valor como explica. “var_prueba” se utiliza para examinar si no se ingreso dato erroneo. En caso de ingresarse un valor valido, se ejecuta el “case” que invoca a un procedimiento dependiendo la opción que se haya elegido.

PROCEDIMIENTO “**ALTA_ENCARGADOS**” (el usuario ingreso opción 1): se utiliza para cargar los datos de los encargados. Primero se pregunta si se quiere alterar el porcentaje de aportes y contribuciones del encargado. Luego se pide el ingreso del DNI, mediante el procedimiento “**validarDNI**” el cual además de pedir el ingreso del DNI ,revisa si el DNI ingresado no es una letra, o si no es un DNI que ya esta utilizando otro encargado ingresando al procedimiento “**búsqueda**” que busca el DNI ingresado en los encargados cargados anteriormente *(es una búsqueda secuencial porque no esta ordenado por DNI el vector)*, si se superan estas 2 pruebas, pregunta si es correcto el DNI ingresado. Luego de salir de este procedimiento se vuelve al procedimiento de alta de encargados continuando así la carga de encargados. Cuando es el momento de cargar el edificio al cual pertenece el encargado, se ingresa al procedimiento “**ValidarEdificio**”. Este procedimiento pide el ingreso del código del edificio para el cual trabaja el encargado (luego de que se haya mostrado la lista de los edificios cargados, tarea que tiene asignada el procedimiento “**ListadoEdificios**”), revisando si el código ingresado no es una letra (dato erróneo) o si no es un código inexistente (el código introducido no pertenece a ningún edificio cargado) utilizando el procedimiento “**BusquedaEdificio**” (*realiza una búsqueda binaria sobre el vector que tiene cargado los edificios, se aprovecha el hecho de estar ordenado por codigo de edificio*). Luego de validar esto se vuelve al procedimiento “alta de encargados” y se continúa la carga de los demás encargados, finalizándola con un 0 (cero) en el DNI, se vuelve así al menú encargados.

PROCEDIMIENTO "**MODIFICAR_ENCARGADO**" (se ingreso opción 2): se utiliza para modificar los datos del encargado y solo se ejecuta si se detecta encargados cargados. En caso de haber encargados cargados y de ejecutarse, primero se muestra una lista de los encargados cargados con el procedimiento "**listado_encargados**" ordenados alfabéticamente (*mas adelante se detalla como funciona este procedimiento*). Luego de mostrar la lista de encargados pedidos se pide el ingreso del DNI del encargado al que se le quiere modificar los datos, se analiza si no se ingreso dato erróneo y luego se

busca el DNI del encargado con el procedimiento "**búsqueda**" que ya se detallo anteriormente. Si no se encuentra el encargado se pide el ingreso de otro mostrando el aviso de que no se lo encontró, en caso de haberse encontrado el encargado se muestra un menú con los campos que representa cada dato del encargado; se ingresa la letra del campo a modificar, y se lo modifica.Si se opta por modificar el edificio para el cual trabaja el encargado, entonces se ingresa al procedimiento "**ValidarEdificio**" procedimiento que ya se explico anteriormente su función. Luego se pregunta si se quiere modificar algún otro campo *(porque solo se puede modificar de a un campo*), en caso negativo se pregunta si se quiere modificar datos de algún otro encargado volviéndose a ejecutar otra vez lo anterior, en caso afirmativo se muestra el menú de los campos otra vez. Se termina de ejecutar este procedimiento cuando el usuario decide no modificar ningún encargado mas.

PROCEDIMIENTO "**BAJA_ENCARGADOS**" (se ingreso opción 3): se utiliza para dar de baja/borrar un encargado. Solo se ejecuta si hay encargados cargados. Al igual que el procedimiento antes explicado, primero muestra el listado de los encargados cargados en orden alfabetico, pide el DNI del encargado a dar de baja, se busca el encargado con el procedimiento "**busqueda**" en caso de haberse encontrado el encargado se pregunta si se quiere dar de baja o no, por las dudas que el usuario se haya equivocado de encargado. Si se decide dar de baja se reemplaza el nombre del encargado a dar de baja por "zzzzzzzz" ,se ordena el vector encargados y se resta 1 al total de encargados, para así mostrar todos los encargados menos el "zzzzzz" (cuando se necesite mostrarlos). En caso de no querer borrar el encargado, se pregunta si se desea dar de baja otro encargado, en caso afirmativo se muestra otra vez la lista y se ejecuta todo otra vez, en caso contrario se vuelve al menú encargados.

PROCEDIMIENTO "**LISTADO_ENCARGADOS**" (se ingreso opción 4 o se lo llamo desde otro procedimiento): se utiliza para mostrar una lista de los encargados cargados junto con los datos cargados de cada uno. Solo se ejecuta si hay encargados cargados. Se ordena primero alfabéticamente utilizando el procedimiento "**ordenaralf**" (*el procedimiento ordenar utiliza un burbujeo optimizado como método de ordenamiento*). Luego de ordenar alfabéticamente se escriben los datos de cada encargado,

(cuando se escribe el apellido hay un conjunto de sentencias que se utilizan para mostrar los datos alineados). Se sale del procedimiento luego de verse la lista y seguir unas instrucciones que solo se tratan de presionar cualquier tecla.

PROCEDIMIENTO "**CARGARDATOSPRUEBA**" (se ingreso opción 5): carga los datos de prueba. Se cargan 3 encargados:

Así se cargan los datos:

DNI	**Apellido y Nombre**	**Edad**	**Sueldo**	**Aport.y contr**	**Edificio**
33123456	Hernandez Jose.	random [30,50]	random [3500,5000]	calcula sobre el ramdom	1
33123457	Baraku Mario.	random [30,50]	random [3500,5000]	calcula sobre el ramdom	2
33123458	Del valle Juan.	random [30,50]	random [3500,5000]	calcula sobre el ramdom	3